

EXHIBIT 8

FILED UNDER SEAL

To: Tad Coburn[Tad.Coburn@sonos.com]; Andrew Schultert[Andrew.Schultert@sonos.com]; Rob Lambourne[Rob.Lambourne@sonos.com]; Joni Hoadley[Joni.Hoadley@sonos.com]; Robert Reimann[Robert.Reimann@sonos.com]; Nick Millington[Nick.Millington@sonos.com]; Jerry Anderson[Jerry.Anderson@sonos.com]; David Taylor[David.Taylor@sonos.com]
From: Ron Kuper[/O=MSTEXCHANGE/OU=SOS ADMIN GROUP/CN=RECIPIENTS/CN=RON.KUPER]
Sent: Fri 7/15/2011 10:09:30 AM Eastern Standard Time
Subject: RE: Play To Sonos
Attachment: PlayToSonos.pdf

Tad asked me to diagram how the play-to-Sonos would work using the IDs thrown over into the cloud. I've made a first pass at this in the picture.

One piece of this is that we want to introduce a new centralized web service for pushing events out to Sonos equipment. The idea is that any Zone or CR that wants to get just-in-time events connects to this service, and then asks to listen for events that match a particular pattern. We would use this service not just for Play-To Sonos, but also to allow us to get just-in-time updates to playlists, starred tracks, etc. (I also envision using this service for any time we want to push events out to our equipment – maybe someday we'd want CS to be able to poke at customer's system if necessary, or we could hook this into our upgrade mechanism somehow.)

Assuming we have this event service, I am envisioning an event called PlayFrom, that takes as parameters (conceptually) the following:

- Service info – the service ID and user name.
- Queue state – could be a single ID or a list of IDs, plus an index and a time offset. I think we need more than just a single ID, because even if the single ID is a playlist we'd want to resume playback from wherever the user left off.
- Target zone group.

There would be a complimentary web service that we'd expose to all content partners, the "Play-To Sonos Service". Our HHs talk to this service to associate Zone Groups with a particular user, and the content partner client talks to this service to enumerate Zone Groups and ask for a Play-To action to be initiated.

Poke holes, please...

-Ron

From: Tad Coburn
Sent: Friday, July 15, 2011 9:05 AM
To: Andrew Schultert; Rob Lambourne; Joni Hoadley; Ron Kuper; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

[AS] Could the ID refer to the third-party app? E.g., could an app point Sonos at a stream that it is generating itself? I would like to think so, and I wonder if that would be a way to implement 1a1 (which I still may not understand).

Tad> Yes, that in fact would be technically possible. If an app already supported the ability to stream audio as, say, a Shoutcast stream, then it might not be too much work to arrange things so the ZP plays that ShoutCast stream. However, for apps that don't already support the ability to stream Audio in a standard format that Sonos supports, I would think there would be little incentive for them to add that functionality to their app just to integrate with Sonos. But I could be wrong.

- Tad

From: Andrew Schultert
Sent: Thursday, July 14, 2011 6:53 PM
To: Tad Coburn; Rob Lambourne; Joni Hoadley; Ron Kuper; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

Hi Tad,

Thanks for the update. Sounds like a good meeting. See some comments below.

-----Original Message-----

From: Tad Coburn
Sent: Thursday, July 14, 2011 6:15 PM
To: Tad Coburn; Rob Lambourne; Joni Hoadley; Ron Kuper; Andrew Schultert; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

Here is one other technical approach which is worth mentioning in the "Play to Sonos" context. This is a variant of the "throw it over the wall" approach that I'll call the "sonos: URI handler" approach.

On many platforms (Android, iOS, Mac, Windows), an application can register a system-wide "URI handler" (a.k.a. "protocol handler") for a given URI scheme (prefix). For instance, the Spotify client app registers to handle any URI that begins with "spotify:". This allows Spotify to do a cool thing: a link to a Spotify playlist can be represented as "spotify:user:topsify:playlist:1QM1qz09ZzsAPiXphF1l4S". This can appear as a link in an email, in a Twitter post, inside a web page, etc. When you click on that link, the system automatically launches the Spotify app and passes the URI to it. The Spotify app then starts playing the playlist.

The Sonos controller app could register to handle all URIs starting with "sonos:" (or something similar). We could define the details of how to create the URI such that a partner could easily form the correct URI to start playing a given ID for that service.

Pros: small amount of work for both partners and Sonos; does require some platform-specific coding for Sonos, but it is small

Cons: requires that the Sonos app be installed on the local device (is this a significant con?)

[AS] I think the biggest con to this approach is that you don't get unsupported platforms (e.g. WinMo) for free. But it's very interesting in that it could provide the simplicity of the cloud approach for, e.g., device discovery without requiring the Internet.

- Tad

-----Original Message-----

From: Tad Coburn

Sent: Thursday, July 14, 2011 5:43 PM

To: Tad Coburn; Rob Lambourne; Joni Hoadley; Ron Kuper; Andrew Schulert; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor

Subject: RE: Play To Sonos

Here is a brief summary of what we discussed at today's meeting on "Play To Sonos" (from my perspective):

1) (from Joni) it is important to make it easy for partners to integrate Sonos into their apps; otherwise they won't do the integration. It is also ok for us to put out a simple initial mechanism and "walk before we run" -- we don't need a gold standard solution for v1 of this feature.

2) There are two general technical approaches for getting the music to Sonos:

- 2a) implement an "audio device driver" for each OS/platform that allows the audio samples to be sent to a ZonePlayer. This would work like a Sonos DOCK in streaming mode. No work for partners; a lot of work for Sonos for each operating system/platform.

[AS] seems like this is something we'd prefer to do in conjunction with an OS partner -- e.g. get Google to implement a SW-only "Airplay".

1a1) Nick mentioned an alternative which is to do the audio streaming at the app layer. This requires deep integration between Sonos and the partner's app and would almost certainly require Sonos delivering a library of code that the partner could integrate into their app. I personally think this is too expensive to consider at this time.

- 2b) the "throw a track (or other playable object) over the wall to Sonos" feature. This means passing the service-specific ID for a playable item (track, playlist, artist, programmed radio station, etc.) to Sonos and telling Sonos to either add the item to the queue or play it now (only the latter is available for programmed radio and streams).

[AS] Could the ID refer to the third-party app? E.g., could an app point Sonos at a stream that it is generating itself? I would like to think so, and I wonder if that would be a way to implement 1a1 (which I still may not understand).

There are two different possibilities for queue management. One is to give the app access to the Sonos queue. The other is for the app to override the Sonos queue with its own app-specific queue. I expect the second is what most third-parties would prefer; we could consider implementing it as an extension to programmed radio - imagine if Pandora let us display previous and following tracks and, instead of displaying "Queue (not in use)" we displayed those tracks.

Comparison:

2a) allows volume control, play/pause and works with ANY app that runs on the given platform/OS. However, there is a pretty high cost to Sonos to implement this, and it may be impossible (unsupported) to implement this approach on some mobile devices. This solution requires moderate bandwidth from a mobile device to the ZPs.

2b) has a lower low cost to implement for Sonos and the cost is not multiplied per platform/OS. We'd expose some web APIs that any partner could use to access the functionality.

Functionality would include:

"list zone groups for user"

"set current zone group for user"

"play id <ID> now on current zone group for user"

(optionally starting at track <trackID>, e.g. for Spotify playlists)

(optionally starting at time offset <MIN:SEC.MS>)

"add id <ID> to the queue on the current zone group for current user"

note: we could add additional web APIs to support remote control of volume and play/pause/skip;

it would be up to the third-party apps to decide whether they want to support these features.

[AS] I'm fine with a crawl/walk/run, and even in "run" mode I think the app should have flexibility in how much it wants to support. But I do think volume control is a "must have" for an initial release.

3) To support device discovery and support setup and configuration, a Sonos web service would act as the intermediary between mobile devices and the Sonos system ZPs. This has a cool upside of allowing a user to queue up music while away from their house (marketing could go to town with this!) IFF we use the "throw it over the wall" approach. Downside: if your internet connection is down, you can't use the "Play To Sonos" feature even from within your own home.

[AS] I'm very intrigued by the idea of using the cloud to locate your ZPs. I don't like requiring the Internet, but I do like getting rid of UPnP device discovery. Is control of your Sonos from outside your house really a feature? Are there other reasons for sending all operations via the cloud? I guess if you're not storing the ZP IP address you don't have to worry about it going stale. And you don't have to worry about how to handle the case where you're not in the right subnet -- it will work regardless (but do you want it to work in that case?). I'm still on the fence here.

4) I think we should not try to define/control the end user experience in other apps very tightly; we can define some possible sample flows, and we should define a standard Sonos "Play To" icon, but we should not define/require specific popup menus, nor demand that the third-party app support Sonos volume and transport control. The idea is to make it as easy as possible for third-parties to integrate this functionality into their app.

[AS] I agree.

5) one open issue: Setup/configuration. How does the enduser "connect" their app to their Sonos system (in a reliable and secure way)? I think the third-party app will need to ask the user to enter their Sonos customer email address and password. The app can then make a request to a Sonos server in the cloud to determine the ZoneGroups on which music can be played.

[AS] This would show all ZoneGroups in all households for that user?

- Tad

p.s. Ron, Devon and I have been discussing a possible server-to-client "near real time" event/notification system which could be a key part of any approach here. It would allow the Sonos server to send an event to a ZP in near-real-time to tell it to start playing a track/playlist/etc and could also be used to send volume change commands and transport control commands.

-----Original Message-----

From: Tad Coburn

Sent: Thursday, July 14, 2011 12:14 PM

To: Rob Lambourne; Joni Hoadley; Ron Kuper; Andrew Schulert; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor

Subject: RE: Play To Sonos

Thanks for adding me to this thread! I think I understand most everyone's points here, and I have a lot of thoughts both on the user experience and possible implementations.

Here are some questions that I'd like everyone to answer to help bound this problem:

- 1) How important is it that it be "easy" for other companies to integrate "Play to Sonos" into their app?
Do we think it should take < 1 person/week to code up? < 1 person/month? < 3 person/months?
- 2) Are we only interested in a "gold standard" user experience when an end user is using some other app to control their Sonos system, or are we interested in a good/better/best hierarchy of features/usability?

The answers to these questions will affect the feedback I give, so I will hold further comments until I hear the answers to the above two questions.

Thanks!

- Tad

-----Original Message-----

From: Rob Lambourne

Sent: Tuesday, July 12, 2011 6:29 PM

To: Rob Lambourne; Joni Hoadley; Ron Kuper; Andrew Schulert; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor; Tad Coburn

Subject: RE: Play To Sonos

Really adding Tad...

Rob Lambourne | User Experience Design | Sonos

-----Original Message-----

From: Rob Lambourne

Sent: Tuesday, July 12, 2011 3:29 PM

To: Joni Hoadley; Ron Kuper; Andrew Schulert; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor

Subject: RE: Play To Sonos

Including Tad,

Tad, this thread is rather lengthy, but I've include the word document I put together as a start (and Joni's response doc).

I think we're in general agreement on what we think is the right end-user experience (perhaps with some debate around volume control), and are discussing the positioning and how this feature relates to the Sonos Controllers.

Rob

Rob Lambourne | User Experience Design | Sonos

I guess I'd characterize/phrase the fourth concern a little differently. I believe one of our primary differentiators / secret sauce is the simple, integrated experience that the controllers provide (all music from all your sources through a single consistent UI). I agree with the comments that users who predominantly or exclusively use a single service in a small household would likely benefit from Play To Sonos (if it were available in a handheld device app, not just the service's desktop client). However, I don't believe that Sonos-centric / multi-zone households will benefit from or care about this feature except in edge cases. For users with more than one zone or who regularly use more than one music service, Play to Sonos would (IMO) provide an inferior user experience in most cases.

I do therefore have a small concern that users introduced to Sonos via the Play to Sonos feature may never adopt using a controller, even once it is in their interest to do so (i.e., they buy more zones or add more services).

So, I think we simply need to keep the use cases in mind and keep the relationship to the controllers in perspective as we move forward with this feature, and provide ways to lead users in the right direction to best serve their needs.

Robert.

-----Original Message-----

From: Joni Hoadley
Sent: Monday, July 11, 2011 2:40 AM
To: Ron Kuper; Andrew Schulert; Rob Lambourne; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

Rob, thanks for putting this together. I think you've captured the majority of use cases here but, like Andy, I thought of a couple more things. I had quite a few comments on your doc, so I just edited it and red-lined it so you could see my comments and questions. Who from PD is going to drive the technical implementation options of this?

Joni

-----Original Message-----

From: Ron Kuper
Sent: Sunday, July 10, 2011 2:55 AM
To: Andrew Schulert; Rob Lambourne; Joni Hoadley; Robert Reimann; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

We should investigate volume control via the O/S, rather than via our Play-To. Both Windows and Mac OS have extensible mixer control panels, I'd guess iOS does, don't know about Android. If we conceptually frame Sonos as a "speaker", then we should control the speaker using whatever UX idiom is already familiar on that platform.

As a general design concept, I think we should consider giving 3rd party apps the ability to sync their state into the cloud, using web services and representation that is like a Sonos queue + what's playing now. Then the Sonos system can grab that state and load the queue from what's in the cloud. That to me seems to neatly address the issues of metadata, pre-buffering, etc.

-Ron

-----Original Message-----

From: Andrew Schulert
Sent: Saturday, July 09, 2011 7:40 PM
To: Rob Lambourne; Joni Hoadley; Robert Reimann; Ron Kuper; Nick Millington; Jerry Anderson; David Taylor
Subject: RE: Play To Sonos

Nice writeup. I have a few comments.

I don't like your fourth reason why Sonos might not want to do this. It sounds like "we shouldn't do this because customers might prefer it." That's of course a reason for doing it, not for not doing it. I do think that this makes us think about the relative merits and positioning of the Sonos controller vs. apps with Send to Sonos. I've come up with the following areas where I expect we'll differentiate for the indefinite future:

- setup and configuration
- simplest way to get started with Sonos and make sense of all music sources out there
- access to music sources that don't support send-to-sonos.
- cross-service features like search and x-reference
- enabling music editorializing outside of a music service app and mapping to music service tracks (e.g., pull track metadata out of a twitter feed and resolve to a track in whichever music service you happen to subscribe to.)

In terms of the user experience from within the third party app, I think you have it exactly right. The absolute key here is to make the extension to the app's UX as simple and minimal as possible. All we are doing (from a user's point of view) is moving the audio from the local speakers to a Sonos zone group. Volume management should work as before, only it will control Sonos, not local speakers.

We could allow additional operations from the 3rd party apps, such as controlling volume or EQ on individual zones, or changing zone grouping. I'm not opposed to this, but I view it as secondary to nailing the simple model above.

The interesting UX questions are in how much control we allow through the Sonos controllers, how that affects the Sonos UX, and how that affects the third-party UX. Here are a list of things we could let you do from Sonos, in rough order of priority:

- break the connection between the app and Sonos (i.e., start playing a different source in the zone group)
- change the volume (should be reflected in the 3rd party UI)
- play/pause
- next/prev track
- Skip directly to a different track in the queue (if we allow the 3rd party app to expose its queue to Sonos)

The technical challenges are really around optimizing the implementation without changing the user model. The user always thinks about it as an Airplay like (or VLO-like) model: we're sending the sound directly from the app to Sonos. If we put this in the OS then that's probably how we'll implement it. But if

we add it to an app then we can take advantage of the specific situation to do things like:

- stream metadata along with the audio, so it appears in the Sonos UI.
- expose the queue, so it appears in the Sonos UI.
- allow audio buffering in the ZonePlayers, to increase robustness.
- stream the audio from the cloud instead of the app.

These are mainly things we already do in one form or another. We need to package them in a way that third-parties can easily take advantage of them.

Andy

-----Original Message-----

From: Rob Lambourne
Sent: Friday, July 08, 2011 11:29 PM
To: Rob Lambourne; Joni Hoadley; Robert Reimann; Ron Kuper; Nick Millington; Jerry Anderson; David Taylor; Andrew Schultert
Subject: RE: Play To Sonos

Adding Andy and David
Rob

-----Original Message-----

From: Rob Lambourne
Sent: Fri 7/8/2011 4:16 PM
To: Joni Hoadley; Robert Reimann; Ron Kuper; Nick Millington; Jerry Anderson
Subject: RE: Play To Sonos

I've made some notes about how the Play-To feature can work from a user perspective. We're not necessarily in agreement on all aspects of this description, but it can form the basis of a discussion next week in the meeting that Joni set up.

<<SONOS PLAY TO functionality.docx>>
Rob

Rob Lambourne | User Experience Design | Sonos

-----Original Appointment-----

From: Joni Hoadley
Sent: Thursday, July 07, 2011 4:26 AM
To: Joni Hoadley; Rob Lambourne; Robert Reimann; Ron Kuper
Cc: Nick Millington; Jerry Anderson
Subject: Play To Sonos
When: Tuesday, July 12, 2011 1:00 PM-2:00 PM (UTC-08:00) Pacific Time (US & Canada).
Where: SB: Rob's office; Cambridge: Ron's office

Let's explore the options for creating code that let's 3rd party apps control Sonos. This is something we want to discuss w/Spotify and would like to get to the point quickly where we can have some wireframes to show our vision for the user experience. I know there are a couple of conflicts but I'm out tomorrow and Monday is completely booked up (looks like Rob's out of the office part of the day) so I'm hoping you can reschedule your other meetings if needed.

Nick/Ron, if there are any thoughts you've captured in writing that you can share ahead of time, that would be very helpful. Thanks!

Joni

Bridge:
Toll free number: 866-299-7945
Guest Code: 1039096